ADVANCES IN MATHEMATICS 50, 160–186 (1983)

A Bijection Proving Orthogonality of the Characters of S_n

DENNIS E. WHITE*

School of Mathematics, University of Minnesota, Minneapolis, Minnesota 55455

Suppose ρ and β are partitions of n. If $\rho \neq \beta$, a bijection is given between positive pairs of rim hook tableaux of the same shape λ and content β and ρ, respectively, and negative pairs of rim hook tableaux of some other shape μ and content β and ρ, respectively. If $\rho = \beta$, the bijection is between positive pairs and either negative pairs or permutations of hooks. The bijection, in the latter case, is a generalization of the Schensted correspondence between pairs of standard tableaux and permutations. If the irreducible characters of S_n are interpreted combinatorially using the Murnaghan–Nakayama formula, these bijections prove

$$\sum_{\lambda} \chi_\rho^\lambda \chi_\beta^\lambda = \delta_{\rho\beta}\, 1^{j_1} j_1!\, 2^{j_2} j_2! \ldots,$$

where $\rho = 1^{j_1} 2^{j_2} \ldots$.

1. INTRODUCTION

In this paper we describe algorithms which give a combinatorial proof of the orthogonality formula

$$\sum_{\lambda \vdash n} \chi_\rho^\lambda \chi_\beta^\lambda = \delta_{\rho\beta}\, 1^{j_1} j_1!\, 2^{j_2} j_2! \ldots.$$

Here, $\lambda \vdash n$ denotes a partition of n and χ_ρ^λ denotes the irreducible character λ of the symmetric group S_n evaluated at an element of type ρ [2]; ρ is a partition of n with j_1 parts of size 1, j_2 parts of size 2,....

The χ_ρ^λ have a well-known combinatorial description using the Murnaghan–Nakayama formula (see [2] or [4]) as the sum of the signs of all the rim hook tableaux of content ρ and shape λ. Thus $\chi_\rho^\lambda \chi_\beta^\lambda$ can be interpreted as pairs of rim hook tableaux of shape λ and content ρ and β, respectively. Each pair will have a sign equal to the product of the signs of the two tableaux.

* Supported in part by a National Science Foundation grant.

0001-8708/83 $7.50

The algorithms in this paper establish:

(1) if $\rho = \beta$, a bijection between positive pairs of rim hook tableaux and either negative pairs of rim hook tableaux or "hook permutations," of which there will be exactly $1^{j_1} j_1! 2^{j_2} j_2! \ldots$.

(2) if $\rho \neq \beta$, a bijection between positive pairs of rim hook tableaux and negative pairs of rim hook tableaux.

Consequences of the construction include a general Schensted algorithm [6] between pairs of rim hook tableaux of content k^m and shape λ and permutations of k-hooks. Many of the properties of the Schensted algorithm carry over to this extension.

In Section 2 we outline the definitions and notation used in this paper. In Section 3 we give the crucial "attack" algorithm, the analog to the Schensted "bumping." Section 4 describes the "insertion" and "deletion" algorithms and Section 5 the "encode" and "decode" algorithms. In Section 6 we state the theorems and corollaries which follow from these algorithms and we make some remarks about further applications.

2. Definitions

A *partition* λ of n (written $\lambda \vdash n$) is a sequence of integers, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ with $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_l \geqslant 1$ and $\lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_l = n$. We say λ has l *parts*. Sometimes we refer to a partition as a *shape* when we interpret it as successive rows of positions or *cells*. Such a diagram is called a *Ferrers diagram* or *Young diagram*.

We sometimes abbreviate the partition λ with the notation $1^{j_1} 2^{j_2} \ldots$, where j_i is the number of parts of size i. Sizes which do not appear are omitted and if $j_i = 1$, it is not written. Thus, $(5, 2, 2, 2, 1)$ can be written $1\, 2^3\, 5$.

Suppose $\mu = (\mu_1, \mu_2, \ldots, \mu_{l_1}) \vdash n_1$ and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{l_2}) \vdash n_2$ and $\mu_1 \geqslant \lambda_1$, $\mu_2 \geqslant \lambda_2, \ldots, \mu_{l_2} \geqslant \lambda_{l_2}$ and $l_1 \geqslant l_2$. The *skew shape* $\mu - \lambda$ is obtained by removing the λ-diagram from the inside of the μ-diagram. (The notation μ/λ is frequently used for $\mu - \lambda$ in the literature. However, in this paper we will usually think of shapes and skew shapes as subsets of cells and for this reason we adopt the set terminology $\mu - \lambda$. We will, in fact, usually refer to skew shapes with a single Greek letter.)

In Fig. 1 we give the shapes $(5, 2, 2, 2, 1)$ and $(6, 6, 3, 1)-(3, 1, 1)$.

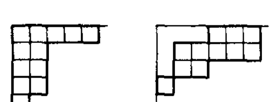


FIGURE 1

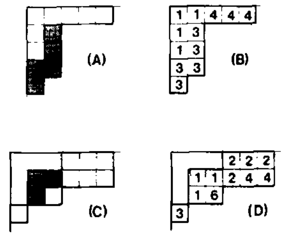


FIGURE 2

A *Composition* ρ of n is a sequence of integers $(\rho_1, \rho_2, ..., \rho_l)$, $\rho_i \geqslant 0$ and $\rho_1 + \rho_2 + \cdots + \rho_l = n$. Note that corresponding to every composition there is a unique partition.

Suppose $\lambda \vdash n$ and ρ is a composition of n. A *tableau* of shape λ and *content* ρ is a Young diagram of shape λ in which the cells have been filled with ρ_1 1's, ρ_2 2's,..., ρ_l l's. We define *skew tableaux* similarly.

A *k-hook* (or *hook*) is a partition $1^i j$, where $i + j = k$. If a hook has a single value in all its cells, it is called a *hook tableau.*

Suppose λ is a shape and α a skew shape where $\alpha = \mu - \lambda'$. The *outer rim* of λ is the set of cells in λ with no cells in λ immediately below and to the right. Similarly define the *inner rim* of α. An *outer rim hook* (or *rim hook*) of λ is a contiguous set of cells in the outer rim of λ whose removal from λ leaves a Young diagram. Similarly define an *inner rim hook* of α.

The *sign* of a rim hook (inner or outer) σ is $\operatorname{sgn}(\sigma) = (-1)^{(\#\text{rows of } \sigma) - 1}$. A *rim hook tableau* of shape λ and content ρ is a tableau such that the ρ_l l's are an inner rim hook of λ and their removal leaves a rim hook tableau. Similarly define a *skew rim hook tableau.* The *sign* of a (skew) rim hook tableau T, $\operatorname{sgn}(T)$, is the product of the signs of its rim hooks.

In Fig. 2(A) is an example of an outer rim 5-hook with sign -1 of $(5, 2, 2, 2, 1)$; in Fig. 2(B) a rim hook tableau of shape $(5, 2, 2, 2, 1)$, content $(4, 0, 5, 3)$ and sign -1; in Fig. 2(C) an inner rim 3-hook of $(6, 6, 3, 1)$–$(3, 1, 1)$ and sign -1; and in Fig. 2(D) a skew rim hook tableau of shape $(6, 6, 3, 1)$–$(3, 1, 1)$, content $(3, 4, 1, 2, 0, 1)$ and sign $+1$.

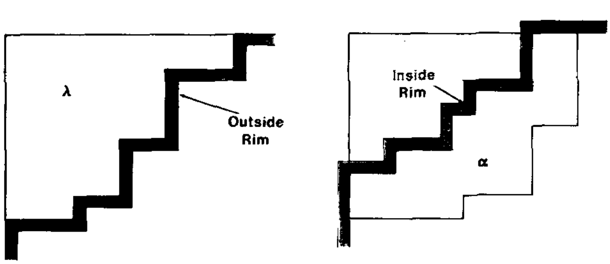


FIGURE 3

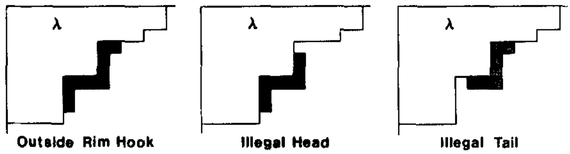


FIGURE 4

The *outside rim* of λ is the set of cells immediately to the right and below λ, or in the first column and below λ, or in the first row and to the right of λ. The *inside rim* of α is the outer rim of λ', plus a row of cells along the top border of λ' starting at the boundary with λ', plus a column of cells along the left border of λ' starting at the boundary with λ'. This 0th column and 0th row are required for proper termination of some of the following algorithms. See Fig. 3. We will call a contiguous set of k cells in the outside rim of λ a *k-snake* or *snake outside* λ. Suppose σ is a snake outside λ. Its *head* is the upper rightmost cell in the snake and its *tail* is the lower leftmost cell. Either σ is an *outside rim hook* of λ, i.e., $\lambda \cup \sigma$ is a shape, or σ has an *illegal head*, an *illegal tail*, or both. See Fig. 4.

We also define *snakes inside* α and *inside rim hooks*. See Fig. 5.

If σ is a snake or rim hook, let $|\sigma|$ denote the number of cells in σ. Suppose σ is a snake outside λ. Let $t \geqslant 0$. Define *slitherup* (λ, σ, t) to be the snake outside λ of length $|\sigma|$ and displaced t cells upward and rightward from σ. Similarly define *slitherdown* (λ, σ, t). See Fig. 6. Similar definitions hold for snakes inside α.

Note that if σ has an illegal head, the tail of slitherup $(\lambda, \sigma, |\sigma|)$ is legal and if σ has an illegal tail, the head of slitherdown $(\lambda, \sigma, |\sigma|)$ is legal.

Let ψ be a collection of cells. Then *bumpout*(ψ) is the collection of cells each of which is immediately below and to the right of a cell in ψ, and *bumpin*(ψ) is the collection immediately above and to the left of ψ. See Fig. 7.

If σ is a rim hook, let $\sigma(j)$ denote the skew rim hook tableau of shape σ with entries all j's. If T is a (skew) rim hook tableau, let $\kappa\langle j\rangle$ or $\kappa_T\langle j\rangle$ denote the rim hook which contains the j's in T. If σ and τ are two rim hooks, define $\sigma_{\text{out}}[\tau] = (\sigma - (\sigma \cap \tau)) \cup \text{bumpout}(\sigma \cap \tau)$.

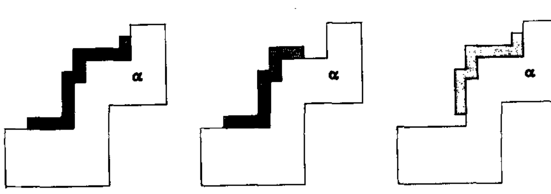


FIGURE 5

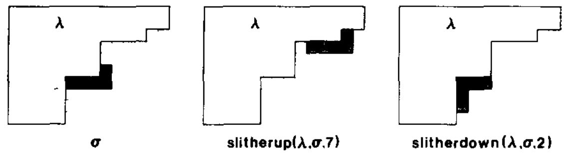


FIGURE 6

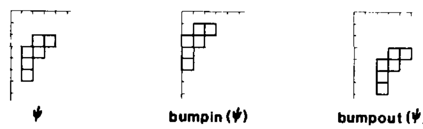


FIGURE 7

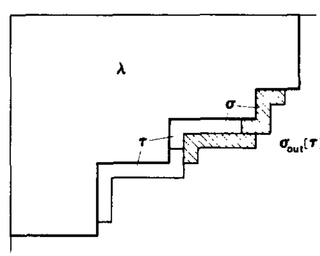


FIGURE 8

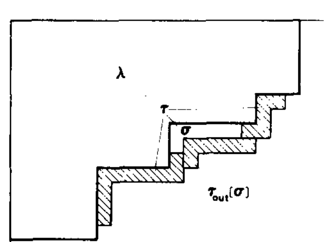


FIGURE 9

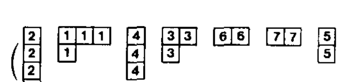


FIGURE 10

LEMMA 1. *If σ and τ are two outside rim hooks of λ such that $\sigma \cap \tau \neq \varnothing$, but $\sigma \nsubseteq \tau$ and $\tau \nsubseteq \sigma$, then $\sigma_{\text{out}}[\tau]$ is an outside rim hook of $\lambda \cup \tau$ and* $\operatorname{sgn}(\sigma_{\text{out}}[\tau]) = -\operatorname{sgn}(\sigma)$.

If σ and τ are two outside rim hooks of λ such that $\sigma \subseteq \tau$ but neither the head nor the tail of τ is in σ, then $\tau_{\text{out}}[\sigma]$ is an outside rim hook of $\lambda \cup \sigma$, bumpout(σ) is an outside rim hook of $\lambda \cup \tau$ and $\operatorname{sgn}(\tau_{\text{out}}[\sigma]) = \operatorname{sgn}(\tau)$.

Proof. See Figs. 8 and 9. In Fig. 8, $\sigma_{\text{out}}[\tau]$ is shaded and in Fig. 9, $\tau_{\text{out}}[\sigma]$ is shaded.

We also define $\sigma_{\text{in}}[\tau] = (\sigma - (\sigma \cap \tau)) \cup \operatorname{bumpin}(\sigma \cap \tau)$. A Lemma similar to Lemma 1 holds for $\sigma_{\text{in}}[\tau]$.

Finally, we say $\mathscr{H} = (H_1, H_2, \ldots, H_m)$ is a *hook permutation* of content $\rho = (\rho_1, \rho_2, \ldots, \rho_m) = 1^{j_1} 2^{j_2} \cdots \vdash n$, and shapes $(\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(m)})$ if the following hold:

(1) each H_i is a hook tableau of shape $\tau^{(i)}$;

(2) $|\tau^{(i)}| = \rho_i$; and

(3) if $\rho_i > \rho_j$, then the content of $H_i <$ content of H_j.

Since the number of hooks of size k is k, the number of hook permutations of content ρ is $1^{j_1} j_1! 2^{j_2} j_2! \ldots$. See Fig. 10 for an example of a hook permutation of type (4, 4, 3, 3, 2, 2, 2).

3. THE ATTACK ALGORITHM

The procedure we describe in this section is the basic building block of the subsequent algorithms. We describe how an area within a shape, called the *attacking hook*, alters two tableaux, one a skew rim hook tableau and the other a rim hook tableau. The result of this procedure is a new attacking hook and two new tableaux.

We describe two such algorithms, one whose movement is generally outward and the other whose movement is generally inward. However, since the two are "mirror images" of one another, we will analyze only one in detail.

The input to this algorithm is a pair of tableaux (T, S) which satisfy Conditions 1–4. The result is a direction (either outward or inward) and a new pair of tableaux $(\hat{T}, \hat{S})$ which also satisfy these four conditions (or their "mirror images," depending upon the direction).

CONDITION 1. T is a rim hook tableau of shape λ with entries $\leqslant j-1$.

CONDITION 2. S is a skew rim hook tableau of shape α with entries $\geqslant j$. We assume j occurs in S.

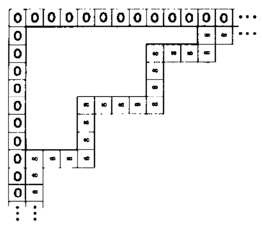


FIGURE 11

CONDITION 3. $\sigma = \lambda \cap \alpha$ is an outer rim hook of λ. (We call σ the *attacking hook.*)

CONDITION 4. σ is an inner rim hook of α.

Note that Conditions 1–3 imply Condition 4 and Conditions 1, 2, and 4 imply Condition 3.

It is convenient, especially for testing for termination, to assume from now on that any (skew) rim hook tableau has ∞ in every cell in its outside rim and 0 in every cell in its left and top borders. See Fig. 11.

Associated with the pair (T, S) is the attacking hook σ, a value j which appears in S and is smallest in S, and a sign $x = \operatorname{sgn}(T)\operatorname{sgn}(S)\operatorname{sgn}(\sigma)$. We let $\tau = \kappa\langle j \rangle$.

Finally, we assume $|\sigma| \leqslant |\tau|$. This gives us four basic cases:

(I) σ and τ are disjoint,

(II) σ and τ overlap,

(III) σ and τ coincide,

(IV) $\sigma \subsetneq \tau$.

Algorithm Attackout (Input: T, S; Output: $\hat{T}$, $\hat{S}$, direction)

begin

if $\sigma \cap \tau = \varnothing$ then

Case I $\quad \begin{cases} \hat{S} \leftarrow S - \tau(j) \\ \hat{T} \leftarrow T \cup \tau(j) \\ \text{direction} \leftarrow \text{outward} \end{cases}$

else if $\sigma \nsubseteq \tau$ then

Case II $\quad \begin{cases} \hat{T} \leftarrow T \cup \tau_{\text{out}}[\sigma](j) \\ \hat{S} \leftarrow S - \tau(j) \\ \text{direction} \leftarrow \text{outward} \end{cases}$

else if $\sigma = \tau$ then

Case III
- $\sigma_2 \leftarrow \sigma$
- repeat
  - $\sigma_2 \leftarrow \text{slitherup}(\lambda - \sigma, \sigma_2, |\sigma_2|)$
- until σ_2 has a legal head
- $\hat{T} \leftarrow T \cup \sigma_2(j)$
- $\hat{S} \leftarrow S - \tau(j)$
- direction $\leftarrow$ outward

else if neither head nor tail of τ is in σ then

Case IV Subcase A
- $\hat{T} \leftarrow T \cup \tau_{\text{out}}[\sigma](j)$
- $\hat{S} \leftarrow S - \tau(j)$
- direction $\leftarrow$ outward

else if head of τ is in σ then

Case IV Subcase B
- $\sigma_2 \leftarrow \text{slitherdown}(\lambda - \sigma, \tau, |\sigma|)$
- if σ_2 has a legal tail on λ then
  - $B(1)$
    - $\hat{T} \leftarrow T \cup \sigma_2(j)$
    - $\hat{S} \leftarrow S - \tau(j)$
    - direction $\leftarrow$ outward
- else
  - B(2)
    - $\sigma_1 \leftarrow \text{bumpin}\ (\sigma_2 - (\tau \cap \sigma_2))$
    - $\hat{S} \leftarrow (S - \sigma(j)) \cup \sigma_1(j)$
    - $\hat{T} \leftarrow T$
    - direction $\leftarrow$ inward

else
- $\sigma_2 \leftarrow \text{slitherup}\ (\lambda - \sigma, \tau, |\sigma|)$
- if σ_2 has legal head on λ then
  - $\hat{T} \leftarrow T \cup \sigma_2(j)$
  - $\hat{S} \leftarrow S - \tau(j)$
  - direction $\leftarrow$ outward
- else
  - $\sigma_1 \leftarrow \text{bumpin}\ (\sigma_2 - (\tau \cap \sigma_2))$
  - $\hat{T} \leftarrow T$
  - $\hat{S} \leftarrow (S - \sigma(j)) \cup \sigma_1(j)$
  - direction $\leftarrow$ inward

end.

We now verify that in each case Conditions 1–4 are maintained. Furthermore, we note any changes in direction and in x. We will denote the shape of $\hat{T}$ by $\hat{\lambda}$, the shape of $\hat{S}$ by $\hat{\alpha}$ and the new attacking hook by $\hat{\sigma}$. Verification is most easily accomplished by careful analysis of the accompanying figure. In each figure, $\hat{\sigma}$ will be shaded, the boundaries of λ and α will be indicated in heavy outline (thus clearly showing σ), and τ will be marked in lighter outline.

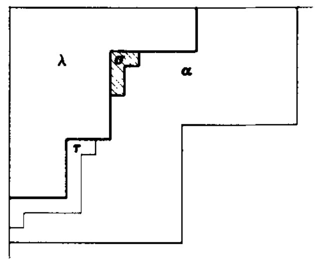


FIGURE 12

First, we note that in every case (except Subcase IVB(2)), $\hat{S} = S - \tau(j)$ so that $\hat{S}$ is automatically a skew rim hook tableau. Thus, we need only show:

(1) $\hat{T}$ is a rim hook tableaux, and

(2) $\hat{\sigma}$ is an outer rim hook of $\hat{\lambda}$.

Case I

See Fig. 12.

(1) Since σ and τ are disjoint, τ is an outside rim hook of λ.

(2) $\hat{\sigma} = \sigma$ which is an outer rim hook of λ and $\lambda \cup \tau$.

Also, $\operatorname{sgn}(\hat{S}) = \operatorname{sgn}(S)\operatorname{sgn}(\tau)$, $\operatorname{sgn}(\hat{T}) = \operatorname{sgn}(T)\operatorname{sgn}(\tau)$, and $\operatorname{sgn}(\hat{\sigma}) = \operatorname{sgn}(\sigma)$ so that $\operatorname{sgn}(\hat{T})\operatorname{sgn}(\hat{S})\operatorname{sgn}(\hat{\sigma}) = x$.

Case II

See Fig. 13.

(1) By Lemma 1, $\tau_{\text{out}}[\sigma]$ is an outside rim hook of λ.

(2) $\hat{\sigma} = \sigma_{\text{out}}[\tau]$. Thus, by Lemma 1, $\hat{\sigma}$ is an outside rim hook of $(\lambda - \sigma) \cup \tau = \hat{\lambda} - \sigma_{\text{out}}[\tau]$.

Also, we have $\operatorname{sgn}(\hat{T}) = \operatorname{sgn}(T)\operatorname{sgn}(\tau_{\text{out}}[\sigma]) = -\operatorname{sgn}(T)\operatorname{sgn}(\tau)$ by Lemma 1, $\operatorname{sgn}(\hat{S}) = \operatorname{sgn}(S)\operatorname{sgn}(\tau)$, and $\operatorname{sgn}(\hat{\sigma}) = \operatorname{sgn}(\sigma_{\text{out}}[\tau]) = -\operatorname{sgn}(\sigma)$ by Lemma 1, so that $\operatorname{sgn}(\hat{S})\operatorname{sgn}(\hat{T})\operatorname{sgn}(\hat{\sigma}) = x$.

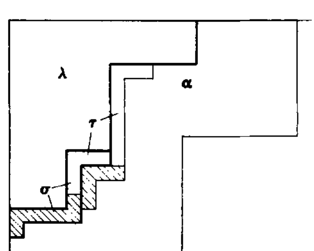


FIGURE 13

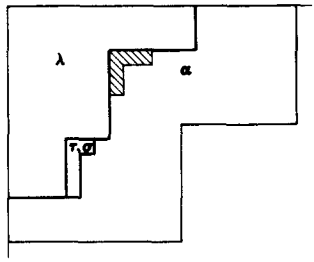


FIGURE 14

Case III

See Fig. 14.

(1) We must show that σ_2 is an outside rim hook of λ. The head of σ_2 is legal on $\lambda - \sigma \Rightarrow$ it is legal on λ. Suppose the tail is illegal on $\lambda - \sigma$. Then the head of the previous σ_2 would be legal on $\lambda - \sigma$. Thus, either the slitherup step would have stopped at this previous step, or the previous σ_2 is σ. The former case contradicts the construction of σ_2, while in the latter case, the head of σ must be to the left of the tail of $\sigma_2 \Rightarrow \sigma_2$ has a legal tail on $(\lambda - \sigma) \cup \sigma = \lambda$.

(2) Since $\hat{\sigma} = \sigma_2$ and $\hat{\lambda} = \lambda \cup \sigma_2$, the same argument applies.

Also, we have $\operatorname{sgn}(\hat{T}) = \operatorname{sgn}(T) \operatorname{sgn}(\hat{\sigma})$, $\operatorname{sgn}(\hat{S}) = \operatorname{sgn}(S) \operatorname{sgn}(\sigma)$, so that $\operatorname{sgn}(\hat{T}) \operatorname{sgn}(\hat{S}) \operatorname{sgn}(\hat{\sigma}) = x$.

Case IV

Subcase A.
See Fig. 15.

(1) Since τ and σ are outside rim hooks of $\lambda - \sigma$, by Lemma 1, $\tau_{\text{out}}[\sigma]$ is an outside rim hook of λ.

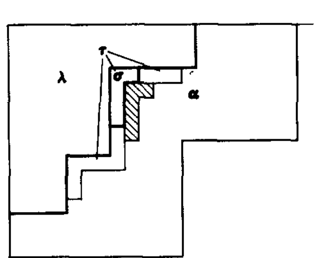


FIGURE 15

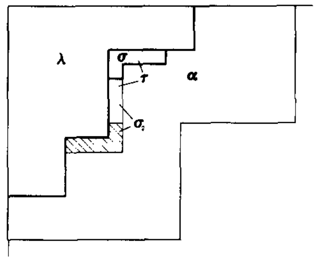


FIGURE 16

(2) $\hat{\sigma} = \text{bumpout}(\sigma)$ so by Lemma 1, it is an outside rim hook of $\lambda \cup \tau \Rightarrow$ it is an outer rim hook of $\hat{\lambda}$.

Also by Lemma 1, $\text{sgn}(\tau_{\text{out}}[\sigma]) = \text{sgn}(\tau)$, so that $\text{sgn}(\hat{T}) = \text{sgn}(T)\,\text{sgn}(\tau)$. But $\text{sgn}(\hat{S}) = \text{sgn}(S)\,\text{sgn}(\tau)$ and $\text{sgn}(\hat{\sigma}) = \text{sgn}(\sigma)$. Thus, $\text{sgn}(\hat{T})\,\text{sgn}(\hat{S})\,\text{sgn}(\hat{\sigma}) = x$.

Subcase B(1).
See Fig. 16.
We consider only the case when the head of τ overlaps σ.

(1) We must show that σ_2 has a legal head and a legal tail outside λ. But the head of σ_2 is below the tail of σ and σ is an inner rim hook of λ, and the tail of σ_2 is legal on $\lambda - \sigma$.

(2) $\hat{\sigma} = \sigma_2 - (\sigma_2 \cap \tau)$. Thus, $\hat{\sigma}$ is a snake outside $\lambda - \sigma$. The head of $\hat{\sigma}$ is below the tail of τ, which is a member of $\lambda \cup \tau$, and the tail of $\hat{\sigma} =$ tail of σ_2 which is legal outside $\lambda - \sigma$ and therefore outside $\lambda \cup \tau$. Thus, $\hat{\sigma}$ is an outside rim hook of $= \hat{\lambda} - \hat{\sigma}$.

Also, $\text{sgn}(\hat{T}) = \text{sgn}(T)\,\text{sgn}(\sigma_2)$, $\text{sgn}(\hat{S}) = \text{sgn}(S)\,\text{sgn}(\tau)$, $\text{sgn}(\hat{\sigma})\,\text{sgn}(\tau \cap \sigma_2) = \text{sgn}(\sigma_2)$ and $\text{sgn}(\sigma)\,\text{sgn}(\tau \cap \sigma_2) = \text{sgn}(\tau)$. Thus, $\text{sgn}(\hat{T})\,\text{sgn}(\hat{S})\,\text{sgn}(\hat{\sigma}) = x$.

Subcase B(2). See Fig. 17.
We again only consider the case where the head of τ is in σ. This is perhaps the most important subcase, since it causes a sign reversal and a

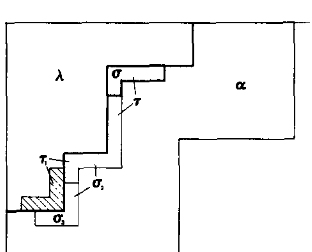


FIGURE 17

change in direction. Since $\hat{T} = T$ is obviously a rim hook tableau, we must show that $\hat{S}$ is a skew rim hook tableau and that σ is an inner rim hook of $\hat{\alpha}$.

Let $\sigma_3 = \sigma_2 - (\tau \cap \sigma_2)$ (which must be a snake), so $\sigma_1 = \text{bumpin}(\sigma_3)$. Let $\tau_1 = (\tau \cap \sigma_2) \cup \sigma_1$. Note that τ_1 is the set of cells containing j's in $\hat{\alpha}$. Note further that $\hat{\alpha} - \tau_1 = \alpha - \tau$, $\hat{\alpha} - \sigma_1 = \alpha - \sigma$, and $\hat{\sigma} = \sigma_1$. We now make three observations:

(1) The head of σ_3 lies directly below the tail of τ, because τ is an outside rim hook of $\lambda - \sigma$. Thus, the head of σ_1 lies to the left of the tail of τ (and the tail of $\tau \cap \sigma_2$), which is a member of $\alpha - \sigma$.

(2) Since the tail of σ_3 (=tail of σ_2) is not legal on $\lambda - \sigma$, it lies to the right of a cell in $\alpha - \sigma$ (or $\alpha - \tau$). Thus the tail of σ_1 (=tail of τ_1) lies above a cell in $\alpha - \sigma$ (or in $\alpha - \tau$).

(3) Since σ is an outside rim hook of $\lambda - \sigma$, the head of τ_1 (=head of $\tau \cap \sigma_2$) lies directly below the tail of σ.

To show S is a skew rim hook tableau, we show τ_1 is an inside rim hook of $\hat{\alpha} - \tau_1$. By observation (1), τ_1 is a contiguous set of cells along the inside rim of $\alpha - \tau$. By observation (2), τ_1 has a legal tail on $\alpha - \tau$. By (3), τ_1 has a legal head on $\alpha - \tau$ (since $\sigma \subseteq \tau$).

Next we show $\hat{\sigma}$ is an inner rim hook of $\hat{\alpha}$. $\hat{\sigma} = \sigma_1$, which is a snake inside $\alpha - \sigma$. By (1), the head of σ_1 is legal inside $\alpha - \sigma$ and by (2), the tail of σ_1 is legal inside $\alpha - \sigma$.

Finally, we have $\text{sgn}(\hat{T}) = \text{sgn}(T)$, $\text{sgn}(\hat{S}) = -\text{sgn}(S)\,\text{sgn}(\sigma)\,\text{sgn}(\hat{\sigma})$, so that $\text{sgn}(\hat{T})\,\text{sgn}(\hat{S})\,\text{sgn}(\hat{\sigma}) = -x$.

We now describe the Attackin Algorithm. This algorithm can be obtained from Attackout by reversing inequalities, replacing slitherup with slitherdown and vice versa, replacing bumpin with bumpout and vice versa, reversing the roles of the skew tableau and the tableau, and replacing inner and inside constructions with outer and outside constructions and vice versa.

The only place where Attackin differs significantly from Attackout is in Case III. This case provides for the only circumstances under which a hook

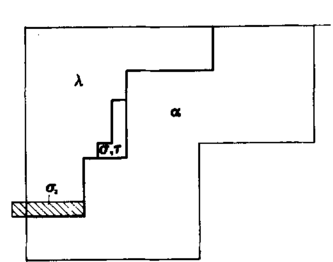


FIGURE 18

can be *bumped out* of the tableau. This coccurs when σ_2 cannot be constructed because slitherdown encounters cells to the left of the first column. See Fig. 18. (Slitherup also encounters border cells in Case IVB(2), but bumpout will position σ_2 correctly.) This special case will stop the deletion algorithm in Section 4 and a hook of j's will be removed from T.

Attackin has two additional outputs: *timetostop*, which is used to indicate when the special circumstances described above occur, and j, the value in τ at the time of this occurrence.

Algorithm Attackin (Input: T, S; Output: $\hat{T}$, $\hat{S}$, j, timetostop, direction)

begin
    timetostop $\leftarrow$ false
    if $\sigma \cap \tau = \varnothing$ then
Case I
        $\hat{S}$—$S \cup \tau(j)$
        $\hat{T} \leftarrow T - \tau(j)$
        direction $\leftarrow$ inward
    else if $\sigma \not\subseteq \tau$ then
Case II
        $\hat{T} \leftarrow T - \tau(j)$
        $\hat{S} \leftarrow S \cup \tau_{in}[\sigma](j)$
        direction $\leftarrow$ inward
    else if $\sigma = \tau$ then
Case III
        $\sigma_2 \leftarrow \sigma$
        repeat
            if slitherdown $(\alpha - \sigma, \sigma_2, |\sigma_2|)$ does not encounter cells to the left of first column
            then
                $\sigma_2 \leftarrow$ slitherdown $(\alpha - \sigma, \sigma_2, |\sigma_2|)$
            else
                timetostop $\leftarrow$ true
        until σ_2 has a legal tail or timetostop
        if timetostop then
            $\hat{T} \leftarrow T - \tau(j)$
            $\hat{S} \leftarrow S$
            $j \leftarrow$ entry in τ
        else
            $T \leftarrow T - \tau(j)$
            $S \leftarrow S \cup \sigma_2(j)$
            direction $\leftarrow$ inward
    else if neither head nor tail of τ is in σ then
Case IV
Subcase A
        $\hat{S} \leftarrow S \cup \tau_{in}[\sigma](j)$
        $\hat{T} \leftarrow T - \tau(j)$
        direction $\leftarrow$ inward

Case IV
Subcase B

else if tail of τ is in σ then
$\sigma_2 \leftarrow$ slitherup $(\alpha - \sigma, \tau, |\sigma|)$
if σ_2 has a legal head on α then
B(1) $\begin{cases} \hat{T} \leftarrow T - \tau(j) \\ \hat{S} \leftarrow S \cup \sigma_2(j) \\ \text{direction} \leftarrow \text{inward} \end{cases}$
else
B(2) $\begin{cases} \sigma_1 \leftarrow \text{bumpout } (\sigma_2 - (\tau \cap \sigma_2)) \\ \hat{T} \leftarrow (T - \sigma(j)) \cup \sigma_1(j) \\ \hat{S} \leftarrow S \\ \text{direction} \leftarrow \text{outward} \end{cases}$
else
$\sigma_2 \leftarrow$ slitherdown $(\alpha - \sigma, \tau, |\sigma|)$
if σ_2 has legal tail on α then
$\begin{cases} \hat{S} \leftarrow S \cup \sigma_2(j) \\ \hat{T} \leftarrow T - \tau(j) \\ \text{direction} \leftarrow \text{inward} \end{cases}$
else
$\begin{cases} \sigma_1 \leftarrow \text{bumpout } (\sigma_2 - (\tau \cap \sigma_2)) \\ \hat{S} \leftarrow S \\ \hat{T} \leftarrow (T - \sigma(j)) \cup \sigma_1(j) \\ \text{direction} \leftarrow \text{outward} \end{cases}$

end.

We make the following crucial observation as a lemma.

LEMMA 2. *Attackout and Attackin are inverse algorithms. That is, the procedure*:

begin
Attackout $(T, S; \hat{T}, \hat{S}, \textit{direction})$
if direction is inward then
Attackout $(\hat{T}, \hat{S}; \hat{\hat{T}}, \hat{\hat{S}}, \textit{direction})$
else
Attackin $(\hat{T}, \hat{S}; \hat{\hat{T}}, \hat{\hat{S}}, j, \textit{timetostop}, \textit{direction})$
end.

yields $T = \hat{\hat{T}}$ *and* $S = \hat{\hat{S}}$ *and timetostop = false; and the procedure*:

begin
Attackin $(T, S; \hat{T}, \hat{S}, j, \textit{timetostop}, \textit{direction})$
if not timetostop then

if direction is inward then
Attackout ($\hat{T}$, $\hat{S}$; $\hat{\hat{T}}$, $\hat{\hat{S}}$, *direction*)
else
Attackin ($\hat{T}$, $\hat{S}$; $\hat{\hat{T}}$, $\hat{\hat{S}}$, j, *timetostop, direction*)

end.

also yields $T = \hat{\hat{T}}$ *and* $S = \hat{\hat{S}}$.

Proof. The proof is immediate since every construction in Attackout is inverted in Attackin.

4. Insertion and Deletion Algorithms

The insertion algorithm is a rim hook analog of the Schensted insertion algorithm [6] for identifying permutations with pairs of standard tableaux. Entire rim hooks sift through the tableau, attacking and "bumping" at various points, according to the rules in Section 3, much as single values do in the Schensted algorithm.

The algorithm given here has as input a rim hook tableau and a hook tableau. The hook tableau must first be positioned so that Attackout in Section 3 can be applied. Suppose λ is a shape and τ a hook.

Algorithm Position (Input: λ, τ; Output: $\hat{\tau}$)

begin
    $\hat{\tau} \leftarrow \tau$
    while $\lambda \cap \hat{\tau} \neq \varnothing$ do
        $\hat{\tau} \leftarrow$ slitherdown (ϕ, $\hat{\tau}$, $|\hat{\tau}|$)
    while $\hat{\tau}$ has an illegal head on λ do
        $\hat{\tau} \leftarrow$ slitherup (λ, $\hat{\tau}$, $|\hat{\tau}|$)
end.

Note that both loops must terminate and the resulting $\hat{\tau}$ is an outside rim hook of λ because an illegal head on $\hat{\tau}$ means a legal tail on slitherup (λ, $\hat{\tau}$, $|\hat{\tau}|$). Let T be a rim hook tableau. We denote by T_j the rim hook tableau obtained by removing all the rim hooks whose content is larger than j (see Fig. 19). We denote the shape of T_{j-1} by λ. Similarly, we denote by T^j

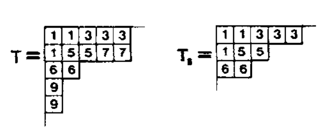


FIGURE 19

the skew rim hook tableau obtained by removing all rim hooks whose content is smaller than j and by α its shape.

Now suppose the shape of T is μ and its content is $(\rho_1, \rho_2, ..., \rho_{j-1}, 0, \rho_{j+1}, ..., \rho_m)$, with $0 \leqslant \rho_i$ for $i \neq j$. Let σ be a hook. We make two assumptions about $|\sigma|$:

ASSUMPTION 1. $|\sigma| \leqslant \rho_i$ for all $i \neq j$.

ASSUMPTION 2. $\rho_i = |\sigma|$ or 0 for all $i > j$.

The point of these assumptions is to cause σ to "attack" only rim hooks of equal length. In Section 6 we shall discuss these assumptions more fully.

The output from the insertion algorithm will be another rim hook tableau $\hat{T}$ with content $\rho = (\rho_1, \rho_2, ..., \rho_m)$, $\rho_j = |\sigma|$, and with shape $\hat{\mu}$ such that $\hat{\sigma} = \hat{\mu} - \mu$ is an outside rim hook of μ. Furthermore, $\mathrm{sgn}(\hat{T}) = \mathrm{sgn}(T)\,\mathrm{sgn}(\hat{\sigma})$.

Algorithm Insert (Input: T, σ, j; Output: $\hat{T}$, $\hat{\sigma}$)

begin

$$
\begin{array}{l}
\text{Position } (\lambda, \sigma; \sigma_1) \\
A \leftarrow T_{j-1} \cup \sigma_1(j) \\
B \leftarrow T^{j+1} \\
\text{while } B \text{ contains finite entries do} \\
\quad \left\{ \begin{array}{l}
\text{Attackout } (A, B; \hat{A}, \hat{B}, \text{direction}) \\
\sigma_{\hat{A}\hat{B}} \leftarrow \text{attacking hook of } \hat{A} \text{ and } \hat{B} \\
A \leftarrow \hat{A} \\
B \leftarrow \hat{B}
\end{array} \right. \\
\hat{T} \leftarrow \hat{A} \\
\hat{\sigma} \leftarrow \sigma_{\hat{A}\hat{B}}
\end{array}
$$

end.

That T has the required properties is a direct result of the discussion of Section 3. Because of Assumptions 1 and 2, no direction reversal occurs and we can use Attackout exclusively. In fact, Case IV of Attackout cannot occur.

At the end, $\hat{A} = \hat{T}$, and $\hat{B}$ contains infinite entries only. Since $\sigma_{\hat{A}\hat{B}}$ is the intersection of $\hat{T}$ and $\hat{B}$, $\sigma_{\hat{A}\hat{B}}$ must be an outside rim hook of μ.

Recall that $\mathrm{sgn}(A)\,\mathrm{sgn}(B)\,\mathrm{sgn}(\sigma_{AB})$ is invariant. At the first step, $\mathrm{sgn}(A) = \mathrm{sgn}(T_{j-1})\,\mathrm{sgn}(\sigma_1)$, $\mathrm{sgn}(B) = \mathrm{sgn}(T^{j+1})$ and the attacking hook σ_{AB} has sign $\mathrm{sgn}(\sigma_1)$. At the end, $\mathrm{sgn}(\hat{A}) = \mathrm{sgn}(\hat{T})$, $\mathrm{sgn}(\hat{B}) = +1$ and $\mathrm{sgn}(\sigma_{\hat{A}\hat{B}}) = \mathrm{sgn}(\hat{\sigma})$. Thus, $\mathrm{sgn}(T) = \mathrm{sgn}(T_{j-1})\,\mathrm{sgn}(T^{j+1}) = \mathrm{sgn}(\hat{T})\,\mathrm{sgn}(\hat{\sigma})$.

We illustrate with an example. T and σ (with j's in the cells of σ) are given in Fig. 20(A). Then Figs. 20(B)–(G) describe A and B (with cells in σ_{AB}

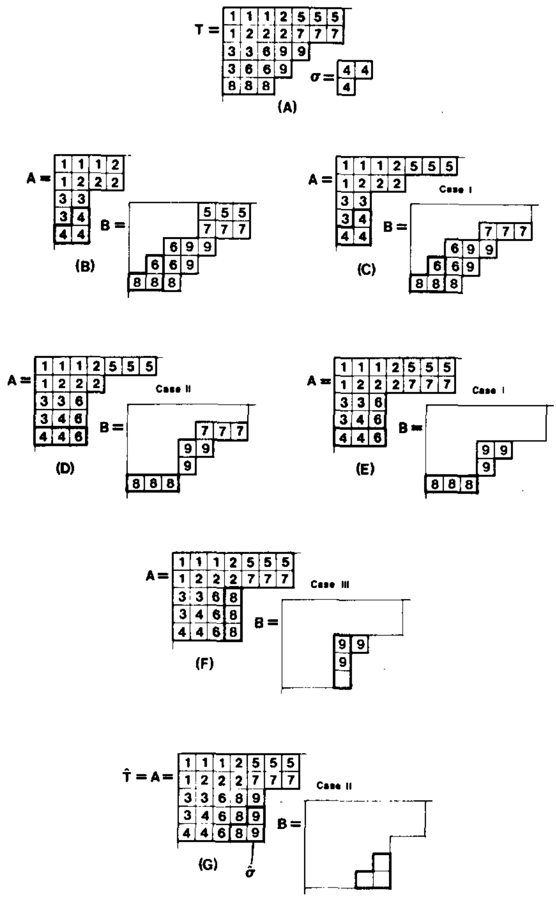


FIGURE 20

marked) at each pass through the main loop. We also give the appropriate case number from Attackout. Note that $\operatorname{sgn}(\hat{T})\operatorname{sgn}(\hat{\sigma}) = -1$ and $\operatorname{sgn}(T) = -1$.

To reverse the insertion algorithm, we use the Attackin Algorithm of Section 3. However, we cannot guarantee that the attacking hook will only encounter hooks of equal size, even if we make the same assumptions about the order of the content of the tableau that we made earlier in this section. However, these encounters with longer rim hooks are exactly what we need to get the required sign reversals and cancellation.

First, we need the following to reverse the Position Algorithm described earlier. Suppose τ is an outside rim hook of λ.

Algorithm Hook (Input: λ, τ; Output: $\hat{\tau}$)

begin

$\hat{\tau} \leftarrow \tau$

repeat

$\hat{\tau} \leftarrow$ slitherdown $(\lambda, \hat{\tau}, |\hat{\tau}|)$

until $\hat{\tau}$ is contained in the first column

repeat

$\hat{\tau} \leftarrow$ slitherup $(\phi, \hat{\tau}, |\hat{\tau}|)$

until $\hat{\tau}$ intersects the first row

end.

We certainly have

LEMMA 3. *Hook and Position are inverses of one another, i.e., the procedure*:

begin

Position $(\lambda, \tau; \hat{\tau})$

Hook $(\lambda, \hat{\tau}; \hat{\hat{\tau}})$

end.

will yield $\tau = \hat{\hat{\tau}}$. *(Reversing the order of Position and Hook in this procedure will also yield* $\tau = \hat{\hat{\tau}}$ *under the special cirumstances that Hook will be used.)*

The deletion algorithm takes a rim hook tableau T of shape μ and content $\rho = (\rho_1, \rho_2, ..., \rho_m)$ and an outer rim hook σ of μ. We require:

ASSUMPTION 3. $|\sigma| \leqslant \rho_i$ for all i.

(We cannot allow the attacking hook to encounter smaller hooks.) There are two possible outcomes:

(1) A rim hook tableau $\hat{T}$ of shape $\hat{\mu}$ and content $\hat{\rho}$; and outer rim hook $\hat{\sigma}$ of $\hat{T}$ such that:

(a) $\hat{\rho} = \rho$;

(b) $\hat{\mu} = (\mu - \sigma) \cup \hat{\sigma}$; and

(c) $\operatorname{sgn}(\hat{T}) \operatorname{sgn}(\hat{\sigma}) = -\operatorname{sgn}(T) \operatorname{sgn}(\sigma)$.

(2) A rim hook tableau $\hat{T}$ of shape $\hat{\mu}$ and content $\hat{\rho}$; a value j; and a hook $\hat{\sigma}$ such that

(a) $\hat{\rho} = (\rho_1, \rho_2, ..., \rho_{j-1}, 0, \rho_{j+1}, ..., \rho_m)$;

(b) $|\rho_j| = |\hat{\sigma}| = |\sigma|$;

(c) $\hat{\mu} = \mu - \sigma$; and

(d) $\operatorname{sgn}(\hat{T}) \operatorname{sgn}(\sigma) = \operatorname{sgn}(T)$.

We differentiate between these cases with the variable *outcome* which takes the values *cancellation* in case (1) and *deletion* in case (2).

Algorithm Delete (Input: T, σ; Output: $\hat{T}$, $\hat{\sigma}$, j, outcome)

```
begin
        A ← T
        B ← σ(∞)
        direction ← inward
        repeat
            { if direction is inward then
                  Attackin (A, B; Â, B̂, j, timetostop, direction)
              else
                  Attackout (A, B; Â, B̂, direction)
              σ_ÂB̂ ← attacking hook of Â and B̂
              A ← Â
              B ← B̂
        until B has no finite entries or timetostop
        if timetostop then
            { T̂ ← Â ∪ B̂
              λ ← shape of Â
              Hook (λ, σ_ÂB̂; σ̂)
              outcome ← deletion
        else
            { T̂ ← Â
              σ̂ ← σ_ÂB̂
              outcome ← cancellation
end.
```

Since sign reversals occur iff direction reversals take place, no pair (A, B) can ever be encountered twice in this algorithm. Thus, it must terminate.

If the outcome is a deletion, the final direction must be inward, so $\operatorname{sgn}(A)$ $\operatorname{sgn}(B)$ $\operatorname{sgn}(\sigma_{AB})$ has changed signs an ever number of times. Thus, $\operatorname{sgn}(\hat{T}) = \operatorname{sgn}(T) \operatorname{sgn}(\sigma)$.

If the outcome is a cancellation, an odd number of direction reversals has occurred, so $\operatorname{sgn}(\hat{T}) \operatorname{sgn}(\hat{\sigma}) = -\operatorname{sgn}(T) \operatorname{sgn}(\sigma)$.

From Lemmas 2 and 3 we have

THEOREM 4. *Insert and Delete are inverses of one another, i.e., the procedure*:

begin

  Delete (T, σ; $\hat{T}$, $\hat{\sigma}$, j, outcome)
  if outcome is cancellation then

Delete $(\hat{T}, \hat{\sigma}; \hat{\hat{T}}, \hat{\hat{\sigma}}, j, outcome)$

else

Insert $(\hat{T}, \hat{\sigma}, j; \hat{\hat{T}}, \hat{\hat{\sigma}})$

end.

yields $T = \hat{\hat{T}}$ *and* $\sigma = \hat{\hat{\sigma}}$, *while the procedure*:

begin

Insert $(T, \sigma, j; \hat{T}, \hat{\sigma})$

Delete $(\hat{T}, \hat{\sigma}; \hat{\hat{T}}, \hat{\hat{\sigma}}, j, outcome)$

end.

yields $T = \hat{\hat{T}}$, $\sigma = \hat{\hat{\sigma}}$ *and the outcome will be deletion.*

Figure 21 gives an example where cancellation occurs. Fig. 21(A) gives T with the cells in σ marked. Figures 21(B)–(G) describe A and B (with σ_{AB}

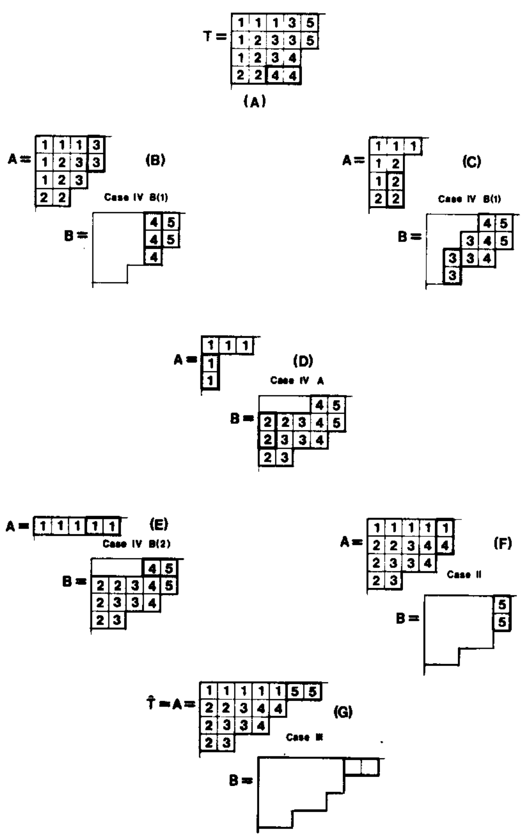


FIGURE 21

marked) at various stages. Note that $\operatorname{sgn}(T)\operatorname{sgn}(\sigma) = +1$ and $\operatorname{sgn}(\hat{T})\operatorname{sgn}(\hat{\sigma}) = -1$. Again case numbers of Attackin and Attackout are included. Case I's are omitted, however.

5. Encoding and Decoding Algorithms

We can now describe an algorithm which assigns to a hook permutation a pair of rim hook tableaux of the same shape and content. Suppose $\rho = (\rho_1, \rho_2, \ldots, \rho_m) \vdash n$. Let $\mathscr{H} = (H_1, \ldots, H_m)$ be a hook permutation of content ρ and shapes $(\tau^{(1)}, \ldots, \tau^{(m)})$.

Algorithm Encode (Input: $\mathscr{H}$; Output: P, Q)

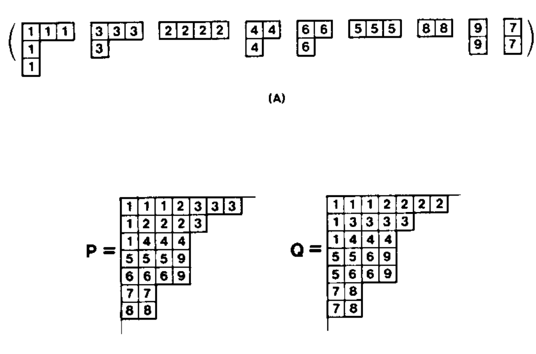
begin

$$
\begin{array}{l}
P, Q \leftarrow \phi \\
\text{for } i \leftarrow 1 \text{ to } m \\
\quad \left\{ \begin{array}{l}
j \leftarrow \text{content of } H_i \\
\text{Insert } (P, \tau^{(i)}, j; \hat{P}, \sigma) \\
\hat{Q} \leftarrow Q \cup \sigma(i) \\
Q \leftarrow \hat{Q} \\
P \leftarrow \hat{P}
\end{array} \right.
\end{array}
$$

end.

In Fig. 22 we give an example of Encode. Figure 22(A) shows $\mathscr{H}$ (of content (5, 4, 4, 3, 3, 3, 2, 2, 2)) and Fig. 22(B) the final P and Q. Note that at each pass through Insert, $\operatorname{sgn}(\hat{P}) = \operatorname{sgn}(P)\operatorname{sgn}(\hat{\sigma})$ and $\operatorname{sgn}(\hat{Q}) = \operatorname{sgn}(Q)\operatorname{sgn}(\hat{\sigma})$. Thus $\operatorname{sgn}(\hat{P})\operatorname{sgn}(\hat{Q}) = \operatorname{sgn}(P)\operatorname{sgn}(Q)$. Since P and Q are initialized to ϕ, $\operatorname{sgn}(P)\operatorname{sgn}(Q) = +1$ at every step and at completion.

The decoding algorithm is defined recursively. Suppose P has content $\rho_1 k_1$'s, $\rho_2 k_2$'s,..., $\rho_m k_m$'s, with $(\rho_1, \ldots, \rho_m) \vdash n$ and $k_1 < k_2 < \cdots < k_m$ and Q



(A)

(B)

Figure 22

has content $\beta = (\beta_1, \ldots, \beta_l) \vdash n$. Furthermore, assume $(\rho_m, \rho_{m-1}, \ldots, \rho_1) \geqslant (\beta_l, \beta_{l-1}, \ldots, \beta_1)$ in lexicographic order, i.e., either $l = m$ and $\beta_m = \rho_m$, $\beta_{m-1} = \rho_{m-1}, \ldots, \beta_1 = \rho_1$ or $\beta_l = \rho_m, \ldots, \beta_{l-i+1} = \rho_{m-i+1}$ and $\beta_{l-i} < \rho_{m-i}$.

The decoding algorithm will produce one of the following two outcomes:

(1) A hook permutation $\mathscr{H} = (H_1, \ldots, H_m)$ of content ρ and shapes $(\tau^{(1)}, \ldots, \tau^{(m)})$. If this result occurs, $l = m$, $\rho_i = \beta_i = |\tau^{(i)}|$ for all i; content of $H_i \in \{k_1, \ldots, k_m\}$; and $\operatorname{sgn}(P) \operatorname{sgn}(Q) = +1$.

(2) A new pair of tableaux $(\hat{P}, \hat{Q})$ with shape $\hat{\mu}$ and content ρ and β, respectively. Furthermore, $\operatorname{sgn}(\hat{P}) \operatorname{sgn}(\hat{Q}) = -\operatorname{sgn}(P) \operatorname{sgn}(Q)$.

We differentiate between the cases with the variable *outcome* which takes values *decoding* in case (1) and *cancellation* in case (2).

Algorithm Decode (Input: P, Q: Output: $\hat{P}$, $\hat{Q}$, $(H_1, \ldots, H_m)$, outcome)

```
begin
        if P and Q both empty then
                outcome ← decoding
        else
        Delete (P, κ_Q⟨l⟩; P̃, σ, j, outcome)
        if outcome is cancellation then
           Case C_1  { Q̂ ← (Q − κ_Q⟨l⟩(l)) ∪ σ(l)
                     { P̂ ← P̃
        else
                Q̃ ← Q − κ_Q⟨l⟩(l)
                Decode (P̃, Q̃; P̃̂, Q̃̂, (H_1,..., H_{m−1}), outcome)
                if outcome is cancellation then
                   Case C_2  { Insert (P̃̂, σ, j; P̂, σ̂)
                             { Q̂ ← Q̃̂ ∪ σ̂(j)
                else
                   Case D    { Outcome ← decoding
                             { H_m ← σ(j)
end.
```

If the outcome is cancellation from case C_1, $\operatorname{sgn}(\hat{P}) = \operatorname{sgn}(\tilde{P})$ and $\operatorname{sgn}(\hat{Q}) = \operatorname{sgn}(Q) \operatorname{sgn}(\kappa_Q\langle l\rangle) \operatorname{sgn}(\sigma)$. But from Delete, $\operatorname{sgn}(P) \operatorname{sgn}(\kappa_Q\langle l\rangle) = -\operatorname{sgn}(\tilde{P}) \operatorname{sgn}(\sigma)$, so that $\operatorname{sgn}(\hat{P}) \operatorname{sgn}(\hat{Q}) = -\operatorname{sgn}(P) \operatorname{sgn}(Q)$.

If the outcome is cancellation from case C_2, $\operatorname{sgn}(\hat{Q}) = \operatorname{sgn}(\hat{\tilde{Q}}) \operatorname{sgn}(\hat{\sigma})$ and from Insert, $\operatorname{sgn}(\hat{P}) = \operatorname{sgn}(\hat{\tilde{P}}) \operatorname{sgn}(\hat{\sigma})$. Thus, $\operatorname{sgn}(\hat{P}) \operatorname{sgn}(\hat{Q}) = \operatorname{sgn}(\hat{\tilde{P}}) \operatorname{sgn}(\hat{\tilde{Q}}) = -\operatorname{sgn}(\tilde{P}) \operatorname{sgn}(\tilde{Q})$ by induction from Decode. But $\operatorname{sgn}(\tilde{Q}) = \operatorname{sgn}(Q) \operatorname{sgn}(\kappa_Q\langle l\rangle)$ and from Delete, $\operatorname{sgn}(\tilde{P}) = \operatorname{sgn}(P) \operatorname{sgn}(\kappa_Q\langle l\rangle)$, giving $\operatorname{sgn}(\tilde{P}) \operatorname{sgn}(\tilde{Q}) = \operatorname{sgn}(P) \operatorname{sgn}(Q)$.

If the outcome is decoding from case D, $\operatorname{sgn}(P) = \operatorname{sgn}(\tilde{P}) \operatorname{sgn}(\kappa_Q\langle l\rangle)$ from

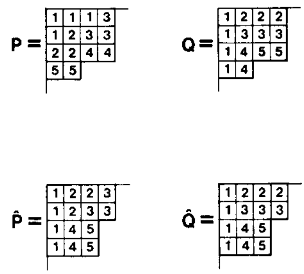


FIGURE 23

delete and $\operatorname{sgn}(Q) = \operatorname{sgn}(\tilde{Q}) \operatorname{sgn}(\kappa_Q\langle l\rangle)$, so $\operatorname{sgn}(P)\operatorname{sgn}(Q) = \operatorname{sgn}(\tilde{P})\operatorname{sgn}(\tilde{Q}) = +1$ by induction.

We illustrate cancellation in Figs. 23 and 24. Figure 23 is an example where Delete produces an immediate cancellation (Case C_1; note the signs of P, Q, $\hat{P}$, and $\hat{Q}$); Fig. 24 is an example where Delete produces a smaller pair and a hook, which cancel in the next call to Decode (case C_2; again, note the signs).

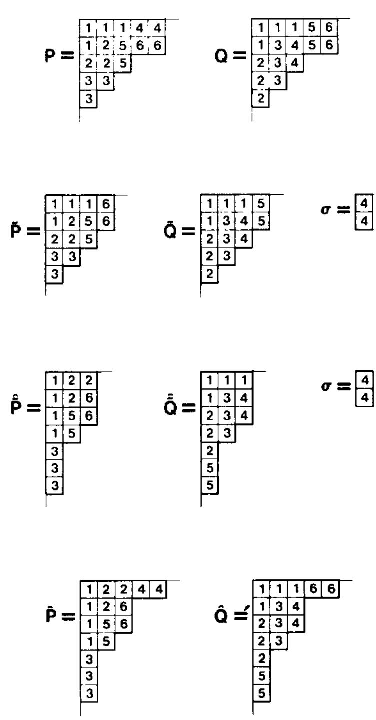


FIGURE 24

We summarize with

THEOREM 5. *Encode and Decode are inverses of one another, i.e., the procedure*:

begin

    Decode $(P, Q; \hat{P}, \hat{Q}, \mathscr{H}, outcome)$
    if outcome is cancellation then
        Decode $(\hat{P}, \hat{Q}; \hat{\hat{P}}, \hat{\hat{Q}}, \mathscr{H}, outcome)$
    else
        Encode $(\mathscr{H}; \hat{\hat{P}}, \hat{\hat{Q}})$

end.

yields $P = \hat{\hat{P}}$ *and* $Q = \hat{\hat{Q}}$, *and the procedure*:

begin

    Encode $(\mathscr{H}; P, Q)$
    Decode $(P, Q; \hat{P}, \hat{Q}, \mathscr{H}, outcome)$

end.

yields $\mathscr{H} = \hat{\mathscr{H}}$ *and outcome = decoding.*

6. THEOREMS, COROLLARIES, AND REMARKS

The algorithms and theorems of Sections 4 and 5 yield the bijection described in the Introduction. Suppose $\rho = (\rho_1, \rho_2, ..., \rho_m) \vdash n$ and $\beta = (\beta_1, \beta_2, ..., \beta_l) \vdash n$ are such that $(\beta_l, \beta_{l-1}, ..., \beta_1) \leqslant (\rho_m, \rho_{m-1}, ..., \rho_1)$ in lexicographic order.

THEOREM 6. *Algorithms Encode and Decode define a bijection between positive pairs* (P, Q) *of rim hook tableaux, where* P *has shape* μ *and content* ρ *and* Q *has the same shape* μ *and content* β, *and*,

(1) *if* $\rho \neq \beta$, *negative pairs of rim hook tableaux* $(\hat{P}, \hat{Q})$, *where* $\hat{P}$ *has shape* $\hat{\mu}$ *and content* ρ *and* $\hat{Q}$ *has the same shape* $\hat{\mu}$ *and content* β, *or*,

(2) *if* $\rho = \beta$, *the union of the set of negative pairs of rim hook tableaux* $(\hat{P}, \hat{Q})$, *where* $\hat{P}$ *has shape* $\hat{\mu}$ *and content* ρ *and* $\hat{Q}$ *has the same shape* $\hat{\mu}$ *and content* β, *with the set of hook permutations of content* ρ.

Proof. This is an immediate consequence of Theorem 5 of Section 5.

THEOREM 7 (Murnaghan–Nakayama formula, [2]). *Suppose* $\lambda \vdash n$. *The* *irreducible character λ of S_n evaluated at an element of S_n of type ρ, χ^λ_ρ, equals*

$$\sum \operatorname{sgn}(T),$$

where the sum is over all rim hook tableaux T of shape λ and content ρ.

COROLLARY 8.

$$\sum_{\lambda \vdash n} \chi^\lambda_\rho \chi^\lambda_\beta = \delta_{\rho\beta} 1^{j_1} j_1 ! 2^{j_2} j_2 !,...,$$

where $\rho = 1^{j_1} 2^{j_2}$....

COROLLARY 9. *Algorithms Encode and Decode construct a bijection between all pairs (P, Q) of k-rim hook tableaux, where P has shape μ and content k^m and Q has the same shape μ and content k^m, and all hook permutations of content k^m.*

Proof. Simply note that if every part of ρ and β is the same size, no sign or direction reversal can take place in Attackin or Attackout.

COROLLARY 10. *Every pair of k-rim hook tableaux (P, Q) of shape μ is positive.*

Proof. If (P, Q) is a negative pair, then Decode must yield a positive pair $(\hat{P}, \hat{Q})$. But applying Decode to $(\hat{P}, \hat{Q})$ gives a hook permutation by Corollary 9 and the original pair (P, Q) by Theorem 5.

We conclude with a few remarks.

Remark 11. Hook permutations of content k^m can be thought of as elements of the wreath product $S_m[Z_k]$ (see [2]) and thus Corollary 9 identifies elements of the wreath product $S_m[Z_k]$ with pairs of k-rim hook tableaux of the same shape and content k^m.

Remark 12. If $k = 2$ in Remark 11, the group is the hyperoctahedral group and the tableaux are "domino tableaux" [8]. In this case, an algorithm of Lustzig [5], using ordinary Schensted, has been used to construct an identification such as that described in Corollary 9.

Remark 13. The Algorithms Insert and Delete bear great resemblance to the Schensted correspondence. In fact, if $k = 1$, they *are* the Schensted correspondence. In a future paper [9] we will show that they share many of the more important features of the Schensted correspondence, including characterizations of inverses, involutions, and inversions (see [3] for a summary of such results). Furthermore, we will give rim hook analogs to the jeu de taquin of Schützenberger [7] and many of its connections to the Schensted correspondence.

Remark 14. It is well known that $\chi^\lambda_\rho = \chi^\lambda_{\rho'}$, where ρ and ρ' are compositions which have the same underlying partition. Combinatorially, this means that

$$\sum \operatorname{sgn}(T) = \sum \operatorname{sgn}(T'),$$

where the left-hand sum is over rim hook tableaux of shape λ and content ρ and the right-hand sum is over rim hook tableaux of shape λ and content ρ'. We have exploited this fact in our Assumptions 1–3 of Section 4, in the requirement that $|\sigma| \leqslant |\tau|$ in Section 3 and in the definition of a hook permutation. We will give a combinatorial proof of this fact, based on the jeu de taquin for rim hooks, in a future paper. This proof will identify

$$\{T : T \text{ is a rim hook tableau of shape } \lambda \text{ and content } \rho \text{ and } \operatorname{sgn}(T) = +1 \text{ or } T \text{ is a rim hook tableau of shape } \lambda \text{ and content } \rho' \text{ and } \operatorname{sgn}(T) = -1\}$$

with

$$\{T : T \text{ is a rim hook tableau of shape } \lambda \text{ and content } \rho \text{ and } \operatorname{sgn}(T) = -1 \text{ or } T \text{ is a rim hook tableau of shape } \lambda \text{ and content } \rho' \text{ and } \operatorname{sgn}(T) = +1\}.$$

Remark 15. Underlying the sign and direction reversals of Delete, Decode, Attackin, and Attackout is the involution principle of Garsia and Milne [1]. We have, in this paper, avoided use of the principle by relying on the assumptions described in Remark 14. However, the combinatorial proof that $\chi^\lambda_\rho = \chi^\lambda_{\rho'}$ mentioned in Remark 14 relies heavily on this principle. Furthermore, some, if not all, the assumptions described in Remark 14 may be dropped in this paper, and the algorithms redegined allowing (1) Insert and Delete to have sign and direction reversals and cancellations and (2) large hooks to attack small hooks. In this situation, the involution principle will play a central role.

REFERENCES

1. A. GARSIA AND S. MILNE, A Rogers–Ramanujan bijection, *J. Combin. Theory Ser. A* **31** (1981), 289–339.
2. G. JAMES AND A. KERBER, The representation theory of the symmetric group, *in* "Encyclopedia of Mathematics and Its Applications" (G.-C. Rota, Ed.), Vol. 16, Addison–Wesley, Cambridge, Mass., 1981.
3. D. KNUTH, *Sorting and Searching*, "The Art of Computer Programming," Vol. 3, Addison–Wesley, Cambridge, Mass., 1973.

4. D. E. LITTLEWOOD, "The Theory of Group Characters," 2nd ed., Oxford Univ. Press, Oxford, 1950.
5. G. LUSZTIG, Communicated by R. Stanley in seminar given at UCSD, 1982.
6. C. SCHENSTED, Longest increasing and decreasing subsequences, *Canad. J. Math.* **13** (1961), 179–191.
7. M. P. SCHÜTZENBERGER, La correspondance de Robinson *in* "Combinatoire et Représentation du Groupe Symétrique," Strasbourg, 1976 (D. Foata, Ed.), 59–113, Lecture Notes in Mathematics, No. 579, Springer-Verlag, Berlin, 1977.
8. R. STANLEY, Some aspects of groups acting on finite posets, *J. Combin. Theory Ser. A* **32** (1982), 132–161.
9. D. STANTON AND D. WHITE, A Schensted algorithm for rim hook tableaux, to appear.